**Project 1**

<Blackjack>

CSC - 5 - 46090

Allende, Katelyn

7/21/2015

Table of Contents

**Introduction 1**

**Summary2**

**Objective 2**

**Variables3**

**C++ Constructs 4**

**Pseudo-Code 5**

**Flowchart 8**

**Program11**

**References 17**

Introduction

Title: Blackjack Game

Blackjack is a popular casino game that is now played all around the world. In the game the aim of the player is to achieve a hand whose points total nearer to 21 than the dealer's hand without exceeding 21.

Equipment-

The game of Blackjack is played with a traditional 52-card deck minus the jokers. Each of these cards are valued at the following:

- An Ace can equal either 1 or 11, whichever makes a better hand.

- Cards 2 from 9 are valued at their face value.

- The 10, Jack , Queen, and King are all valued at 10.

Rules of the Game-

At the start of the game each player starts with two cards. One of the dealer's cards is hidden throughout the entire game until the end. Upon looking at your first two cards the player has two choices. The player can either 'Hit' and receive an additional card or 'Stay' to hold their total and terminate their turn. The dealer must hit until their cards total 17 or higher.

If the player's hand exceeds 21 they bust and the dealer automatically wins no matter the score of their hand. If the player's hand is closer to 21 without exceeding that amount they have beat the dealer. Finally, a blackjack is a total of 21 in their first two cards, which results in an automatic win.

Summary

Project Size: 291 lines

Number of Variables: 27

Number of Methods: 17

* Incorporated functions, random number generator, if-else statements, Boolean statements, arrays, and a variation of loops: do-while, while, and for-loops. I was unsure how to pull from a file for this game.
* This program is a base game and placing bets will be an addition to the following project.
* Completion of the game took approximately a week.
* The difficulty of the project was moderate and made easier after learning how to apply functions.
* It was challenging to program my first game, but rewarding after it ran successfully.

Objective

To program a game using the constructs learned in class. The program allows the player to play the game of Blackjack against the dealer (computer).

Variables

|  |  |  |
| --- | --- | --- |
| Type | Variable Name | Description |
| bool | CrdsDlt[52] | Deck of Cards |
| int | hCrdCnt | House Card Count |
| int | HseHnd[12] | Array of House Hand |
| int | pCrdCnt | Player Card Count |
| int | PlyrHnd[12] | Array of Player Hand |
| bool | doAgain | Loop decision to play again or not |
| char | pChoice | Choice to Hit or Stay |
| bool | pHits | Player Hits |
| int | pScore | Player's Score |
| int | hScore | House's Score |
| bool | hBusts | House Bust |
| char | response | Decision to continue or not |
| const int | hCnt | House Count |
| const int | pCnt | Player Count |
| int | h[] | Hand |
| const int | t | Card Count |
| int | AceCnt | Ace Count |
| int | Score | Score of Hand |
| int | newCrd | New Card |
| int | iRank | Rank of card |
| int | r | Loop to print hand |
| int | newCard | New Card |
| bool | d | Decision for next card or not |
| const int | iNxtCrd | Next Card |
| const int | cRank | Card Rank |
| const int | cSuit | Card Suit |
| int | i | Loop for shuffle |

C++ Constructs

|  |  |
| --- | --- |
| Syntax/Keywords | Location |
| void (functions) | void shuffle, PrntCrd, PrntHnd, PrntSaH |
| int (functions) | int NxtCard, ScrHnd |
| srand | (time(0)) |
| rand () | newCard |
| do-while | doAgain, NxtCard |
| while | (pHits&&pScore<22), (hScore<17), (AceCnt>0&&Score<12) |
| if-else | pChoice, pScore, hBusts, iRank, cRank, cSuit |
| if | CrdsDlt |
| for | ScrHnd, PrntHnd, shuffle |
| true | pHits, doAgain, d |
| false | pHits, doAgain, d, CrdDlt |
| Equality operators and relational operators  (==,!,>,<,&&) | Numerous locations throughout code |
| Arithmetic operators  (+, =) | Numerous locations throughout code |
| Increment operator (++) | pCrdCnt, hCrdCnt, r, AceCnt, Score, i |
| Decrement operator (--) | AceCnt |
| % | iRank, newCard, cRank, cSuit |
| return | 0, Score, newCard |

Pseudo-Code

Declare Variables

Set the random number seed

Do

Shuffle the cards

Deal two cards to each player

Heading for a new hand

Do

Display first two cards

Ask player if they would like to hit or stay

If answer is hit

PlyrHnd[pCrdCnt] equals pull from NxtCard function

Increment operator for pCrdCnt

Else if answer is stay

Hit equals false

Else

Print "Invalid input. Try Again."

Update score, check for bust, and see who won

pScore equals pull from ScrHnd function (PlyrHnd , pCrdCnt)

While ( pHits and pScore is less and 22)

If pScore is greater than 21

Output "THE HOUSE WINS!!!"

Pull from PrntSaH function

Else if pScore equals equals 21

Output "THE PLAYER WINS!!!"

Pull from PrntSaH function

Else

hScore equals pull from ScrHnd function (HseHnd, hCrdCnt)

While hScore is less than 17

HseHnd [hCrdCnt] equals pull from NxtCard function

Increment CrdCnt

hScore equals pull ScrHnd function (HseHnd, hCrdCnt)

hBusts equals hScore is less than 21

If hBusts

Output "THE PLAYER WINS!!!"

Pull from PrntSaH function

Else

If pScore equals equals hScore

Output "TIE!!!"

Pull from PrntSaH function

Else if pScore is greater than hScore

Output "THE PLAYER WINS!!!"

Pull from PrntSaH function

Else

Output "THE HOUSE WINS!!!"

Pull from PrntSaH function

Prompt if they would like to continue

If response equals equals 'y'

doAgain equals true

Else

doAgain equals false

While (doAgain)

Return

Void PrntSaH function

Output ScrHnd function for both the Hse/Plyr Hnd and h/ p Cnt

Int ScrHnd function

Declare Variables

For

Increment r if r is less and t

newCrd equals h[r]

iRank equals newCrd mod 13

If iRank equals equals 0

Increment AceCnt and Score

Else if iRank is less than 9

Score equals score plus (iRank plus 1)

Else

Score equals score plus 10

While AceCnt is greater than 0 and score is less than 12

Decrement AceCnt

Score equals score plus 10

Return Score

Int NxtCard function

Declare Variables

Do

newCard equals random number seed mod 52

If CrdsDlt not [newCard]

d equals false

While (d)

Return newCard

Void PrntHnd

For

Increment r if r is less than t

iNxtCrd= h[r]

Pull from PrntCrd function (iNxtCrd)

Void PrntCrd

Declare Variables

Print the rank of the card

If cRank equals equals 0

Output 'A'

Else if cRank is less than 9

cRank plus 1

Else if cRank equals equals 9

Output 'T'

Else if cRank equals equals 10

Output 'J'

Else if cRank equals equals 11

Output 'Q'

Else

Output 'K'

Print the suit of the card

If cSuit equals equals 0

Output 'C'

Else if cSuit equals equals 1

Output 'D'

Else if cSuit equals equals 2

Output 'H'

Else

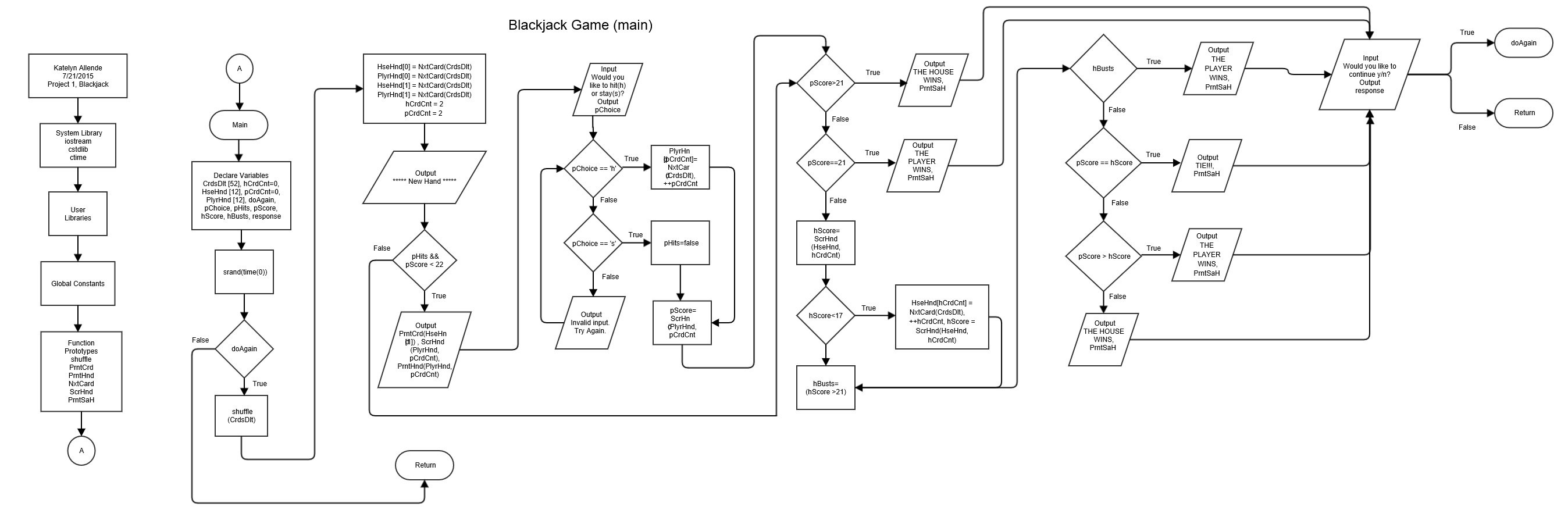
Output 'S'

Void shuffle

For

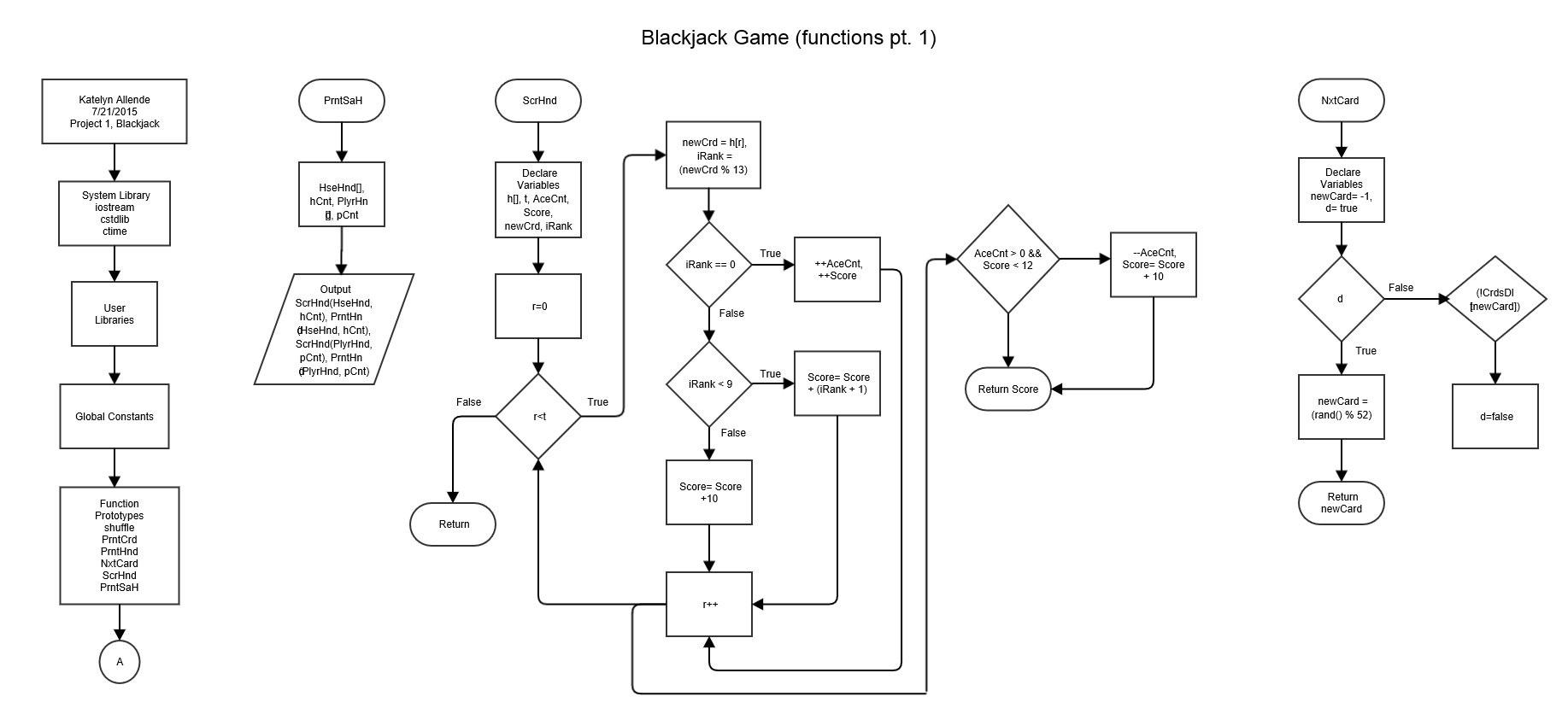
Increment i if i is less than 52

CrdsDlt[i] equals false

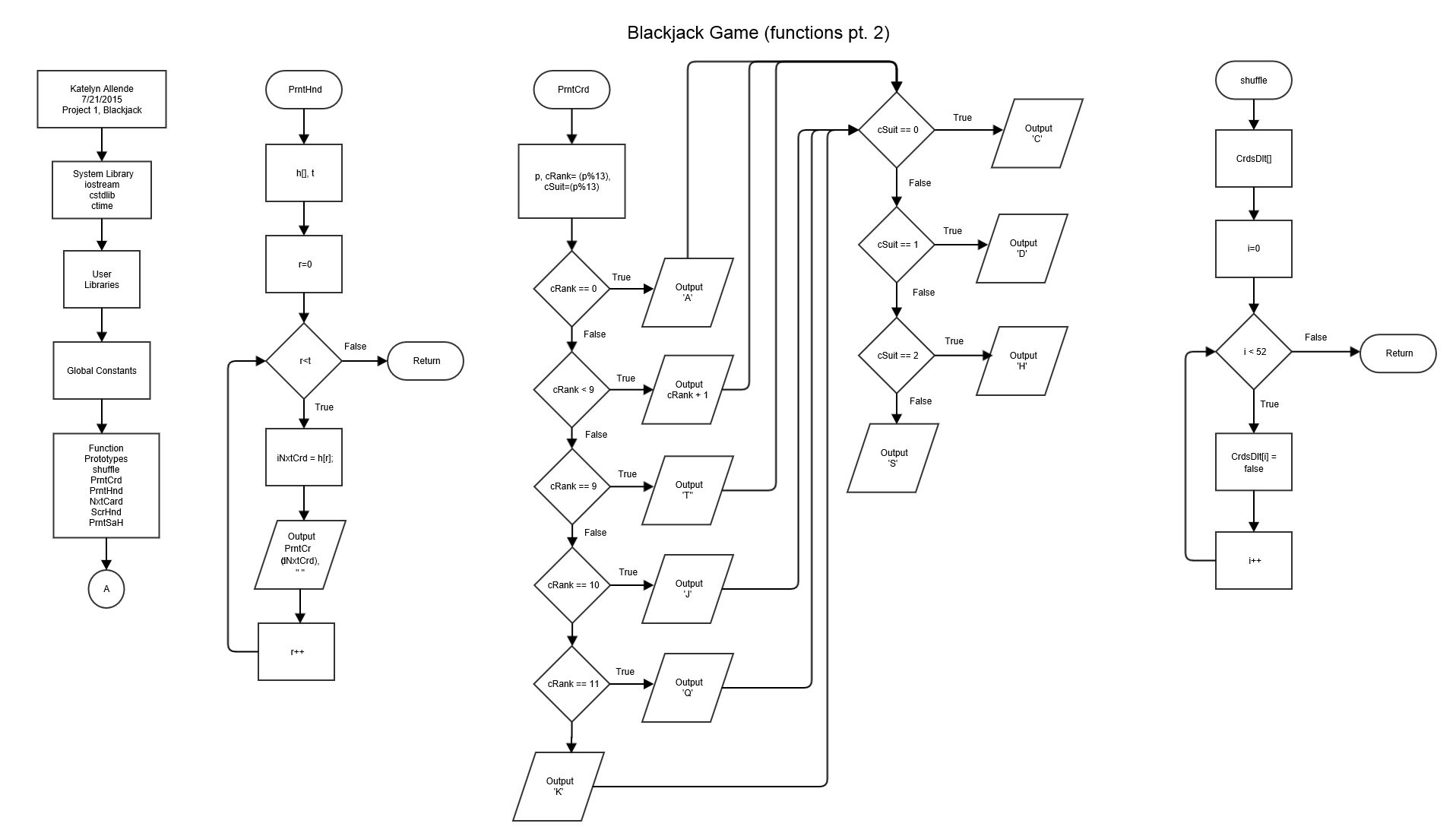
Flowchart (main)

**\*flowcharts additionally in Project folder**

Flowchart (functions pt. 1)



Flowchart (functions pt. 2)



Program

/\*

\* File: main.cpp

\* Author: Katelyn Allende

\* Created on July 17, 2015, 2:10 PM

\* Purpose: Midterm - Create a game -

\* Blackjack

\*/

//System Libraries

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

//User Libraries

//Global Constants

//Function Prototypes

void shuffle(bool []);

void PrntCrd(int);

void PrntHnd(int [], const int);

int NxtCard(bool []);

int ScrHnd(int [], const int);

void PrntSaH(int [], const int, int [], const int);

//Execution Begins Here!

int main(int argc, char\*\* argv) {

//Declare Variables

bool CrdsDlt[52]; //Deck of Cards

int hCrdCnt = 0; //House Card Count

int HseHnd[12]; //Array of House Hand

int pCrdCnt = 0; //Player Card Count

int PlyrHnd[12]; //Array of Player Hand

bool doAgain; //Decision to play again or not

char pChoice; //Choice to Hit or Stay

bool pHits = true; //Player Hits

int pScore; //Player's Score

int hScore; //House's Score

bool hBusts; //House Bust

//Set the random number seed

srand(time(0));

//Loop for each hand

do {

//Shuffle the cards

shuffle(CrdsDlt);

//Deal two cards to each player

HseHnd[0] = NxtCard(CrdsDlt);

PlyrHnd[0] = NxtCard(CrdsDlt);

HseHnd[1] = NxtCard(CrdsDlt);

PlyrHnd[1] = NxtCard(CrdsDlt);

hCrdCnt = 2;

pCrdCnt = 2;

//Heading for a new hand

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* New Hand \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

//Display first two cards

do {

cout << "House's Hand" << endl;

cout << "\*\*";

PrntCrd(HseHnd[1]);

cout << endl;

cout << "Player's Hand Score: " << ScrHnd(PlyrHnd, pCrdCnt) << endl;

PrntHnd(PlyrHnd, pCrdCnt);

//Ask player if they would like to hit or stay

cout << "Would you like to hit(h) or stay(s)?" << endl;

cin >> pChoice;

if (pChoice == 'h') {

PlyrHnd[pCrdCnt] = NxtCard(CrdsDlt);

++pCrdCnt;

} else if (pChoice == 's') {

pHits = false;

} else {

cout << "Invalid input. Try Again." << endl;

}

cout << endl;

//Update Score, Check for Bust, and See Who Wins

pScore = ScrHnd(PlyrHnd, pCrdCnt);

}while (pHits && pScore < 22);

if (pScore > 21) {

cout << "-------------------------" << endl;

cout << "THE HOUSE WINS!!!" << endl;

PrntSaH(HseHnd, hCrdCnt, PlyrHnd, pCrdCnt);

}else if (pScore==21){

cout << "-------------------------" << endl;

cout << "THE PLAYER WINS!!!" << endl;

PrntSaH(HseHnd, hCrdCnt, PlyrHnd, pCrdCnt);

}else {

hScore = ScrHnd(HseHnd, hCrdCnt);

while (hScore < 17) {

HseHnd[hCrdCnt] = NxtCard(CrdsDlt);

++hCrdCnt;

hScore = ScrHnd(HseHnd, hCrdCnt);

}

hBusts = (hScore > 21);

if (hBusts) {

cout << "-------------------------" << endl;

cout << "THE PLAYER WINS!!!" << endl;

PrntSaH(HseHnd, hCrdCnt, PlyrHnd, pCrdCnt);

}else {

if (pScore == hScore) {

cout << "-------------------------" << endl;

cout << "TIE!!!" << endl;

PrntSaH(HseHnd, hCrdCnt, PlyrHnd, pCrdCnt);

}else if (pScore > hScore) {

cout << "-------------------------" << endl;

cout << "THE PLAYER WINS!!!" << endl;

PrntSaH(HseHnd, hCrdCnt, PlyrHnd, pCrdCnt);

}else {

cout << "-------------------------" << endl;

cout << "THE HOUSE WINS!!!" << endl;

PrntSaH(HseHnd, hCrdCnt, PlyrHnd, pCrdCnt);

}

}

}

//Prompt if they would like to continue

cout << endl << "Would you like to continue y/n?" << endl;

char response;

cin >> response;

cout << endl;

if (response == 'y') doAgain = true;

else doAgain = false;

} while (doAgain);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PrntSaH \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: To print the score and hand.

\* Input:

\* HseHnd -> House Hand

\* hCnt -> House Card Count

\* PlyrHnd -> Player Hand

\* pCnt -> Player Card Count

\* Input-Output:

\* ScrHnd -> Score Hand for House and Player

\* PrntHnd -> Print Hand for House and Player

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void PrntSaH(int HseHnd[], const int hCnt, int PlyrHnd[], const int pCnt) {

//Output each score and the hand

cout << "House's Hand Score: " << ScrHnd(HseHnd, hCnt) << endl;

PrntHnd(HseHnd, hCnt);

cout << "Player's Hand Score: " << ScrHnd(PlyrHnd, pCnt) << endl;

PrntHnd(PlyrHnd, pCnt);

cout << endl;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ScrHnd \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: To determine the score of the hand.

\* Input:

\* h -> Hand

\* t -> Card Count

\* Output:

\* score -> Score of hand

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int ScrHnd(int h[], const int t) {

//Declare Variables

int AceCnt = 0; //Ace Count

int Score = 0;

int newCrd; //New Card

int iRank; //Rank of Card

//For-loop to determine score of hand

for (int r = 0; r < t; r++) {

newCrd = h[r];

iRank = (newCrd % 13);

if (iRank == 0) {

++AceCnt;

++Score;

} else if (iRank < 9) {

Score = Score + (iRank + 1);

} else {

Score = Score + 10;

}

}

while (AceCnt > 0 && Score < 12) {

--AceCnt;

Score = Score + 10;

}

return Score;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NxtCrd \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: To deal next card(s).

\* Input:

\* CrdsDlt -> Deck of cards

\* Output:

\* newCard -> Next card(s)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int NxtCard(bool CrdsDlt[]) {

//Declare Variable

int newCard = -1;

bool d = true;

//Loop to get next card

do {

newCard = (rand() % 52);

if (!CrdsDlt[newCard]) {

d = false;

}

} while (d);

return newCard;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PrntHnd \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: To print the card hand.

\* Input:

\* h -> Hand

\* t -> Card Count

\* Input-Output:

\* On Screen

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void PrntHnd(int h[], const int t) {

//For-loop to print hand of cards

for (int r = 0; r < t; r++) {

const int iNxtCrd = h[r];

PrntCrd(iNxtCrd);

cout << " ";

}

cout << endl;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PrntCrd \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: To print the card.

\* Input:

\* p -> Card

\* Input-Output:

\* On Screen

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void PrntCrd(int p) {

//Declare Variables

const int cRank = (p % 13); //Card Rank

const int cSuit = (p % 13); //Card Suit

//Print the rank of the card

if (cRank == 0) {

cout << 'A';

} else if (cRank < 9) {

cout << (cRank + 1);

} else if (cRank == 9) {

cout << 'T';

} else if (cRank == 10) {

cout << 'J';

} else if (cRank == 11) {

cout << 'Q';

} else {

cout << 'K';

}

//Print the suit of the card

if (cSuit == 0) {

cout << 'C';

} else if (cSuit == 1) {

cout << 'D';

} else if (cSuit == 2) {

cout << 'H';

} else {

cout << 'S';

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* shuffle \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: To shuffle the deck of cards.

\* Input:

\* CrdsDlt -> Deck of cards

\* Input-Output:

\* bool CrdsDlt -> Shuffled cards

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void shuffle(bool CrdsDlt[]) {

//For-loop to shuffle cards

for (int i = 0; i < 52; i++) {

CrdsDlt[i] = false;

}

}

References

Gaddis, Tony. *Starting out with C++ : From Control Structures through Objects*. Boston: Pearson, 2015. Print.

https://www.blackjackinfo.com/blackjack-rules/blackjack-basics/